Научно-исследовательский практикум

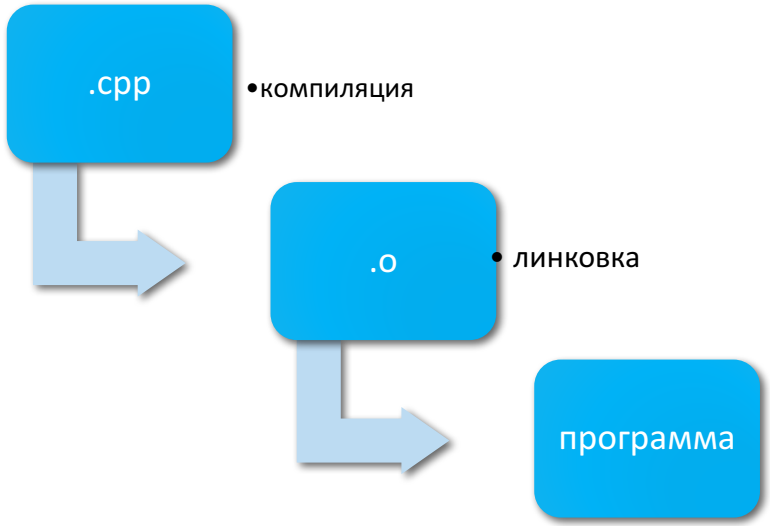
ШАБЛОНЫ. УМНЫЙ УКАЗАТЕЛЬ

Шаблоны: инстанцирование

До этого мы говорили, что нужно разделять реализцию класса на .h и .cpp файлы

При этом .cpp файл представляет собой *единицу трансляции*

Такой подход позволяет минимизировать необходимое при компиляции количество действий



Но с шаблонами всё иначе!

Явное и неявное инстанцирование

Неявное инстанцирование:

Шаблонный класс (функция) находится в заголовочном файле

В этом случае при включение заголовочного файла мы можем подставить любой аргумент

Явное инстанцирование

Мы разделяем класс и его реализацию

Тогда реализация – тоже *единица трансляции*, и нужно сделать явное инстанцирование

Явное инстанцирование

SOMECLASS.H

```
template <typename T>
class SomeClass
{
public:
  T GetData();
  T SetData(T val);
private:
  T data;
};
```

SOMECLASS.CPP

```
template<typename T>
T SomeClass<T>::GetData()
{
   return data;
}

template<typename T>
T SomeClass<T>::SetData(T val)
{
   data = val;
}
```

Программа

```
SomeClass<int> example1;
example1.SetData(1);
cout << example1.GetData() << std::endl;
SomeClass<int> example2;
example2.SetData(3.14);
cout << example2.GetData() << std::endl;
SomeClass<double> example3;
example3.SetData(3.14);
cout << example3.GetData() << std::endl;
```

Результат компиляции

```
Undefined symbols for architecture x86_64:
  "SomeClass<double>::GetData()", referenced from:
      _main in main.o
  "SomeClass<double>::SetData(double)", referenced from:
      _main in main.o
  "SomeClass<int>::GetData()", referenced from:
      _main in main.o
  "SomeClass<int>::SetData(int)", referenced from:
      _main in main.o
ld: symbol(s) not found for architecture x86_64
clang: error: linker command failed with exit code 1 (use -v to see invocation)
```

Инстанци-рование

Изменим someclass.cpp

```
template<typename T>
T SomeClass<T>::GetData()
{
    return data;
}

template<typename T>
T SomeClass<T>::SetData(T val)
{
  data = val;
}

template class SomeClass<int>;
template class SomeClass<double>;
```

Результат

```
Starting /Users/amakashov/projects/build-static_elements
1
3
3.14
/Users/amakashov/projects/build-static_elements-Desktop-
```

Неявное инстанцирование – всё в someclass.h

```
template <typename T>
class SomeClass
{
public:
     T GetData() {return data;}
     T SetData(T val) {data = val;}
private:
     T data;
};
```

Статические функции и шаблоны

Изменим слегка наш класс:

```
template <typename T>
class SomeClass
{
      …
      static int TotalCount() {return count;}
private:
      static int count;
}
```

Также нам нужна инициализация переменной

В someclass.cpp

```
template <typename T>
int SomeClass<T>::count = 0;
```

main.cpp

```
SomeClass<int> example1;
example1.SetData(1);
cout << example1.GetData() << std::endl;
SomeClass<int> example2;
example2.SetData(3.14);
cout << example2.GetData() << std::endl;
SomeClass<double> example3;
example3.SetData(3.14);
cout << example3.GetData() << std::endl;
cout<< "Int classes " << SomeClass<int>::TotalCount() << endl;
cout<< "Int classes " << SomeClass<double>::TotalCount() << endl;
```

Результат

```
Starting /Users/amakashov/projects/build-static_elements-Desktop-De
1
3
3.14
Int classes2
Int classes1
/Users/amakashov/projects/build-static_elements-Desktop-Debug/stati
```

Умный указатель

Идея – сделать указатель произвольного типа

Но такой, чтобы сам собой удалялся

Как – сделать шаблонны класс (и что нибудб перегрузить)

Реализация

```
template <typename T>
class SmartPointer
{
public:
SmartPointer()
{
    m_pointer = new T;
}

~SmartPointer()
{
    if (m_pointer)
       delete m_pointer;
}

T& operator* () const
{
    return *m_pointer;
}

T* operator->() const
{
   return m_pointer;
}

protected:
   T* m_pointer;
};
```

main.cpp

```
SmartPointer<double> doublePtr;
SmartPointer<TestClass> classPtr;

*doublePtr = 10;
cout << "Value is " << *doublePtr << endl;

classPtr->SomeFunction1();
classPtr->SomeFunction2();
```

Результат

```
Starting /Users/amakashov/projects/build-static_elements-
Value is 10
Text from some function
Text from other function
/Users/amakashov/projects/build-static_elements-Desktop-D
```